

Response to Final Office Action
Docket No. 011.0201.US.UTLAmendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (currently amended): A system for efficiently forwarding client requests in a distributed computing environment, comprising:
 - 3 a socket receiving a plurality of non-proxyable client requests commonly addressed routed for forwarding to an origin server from individual sending clients into a proxy server connectively interposed between the sending clients and the origin server;
 - 7 a time estimates generator dynamically generating at the proxy server, concurrent to and during processing of each request by the proxy server, time estimates of service availability based on a time-to-idle for sending the requests over each of a plurality of network connections from the proxy server to the origin server, wherein time-to-idle for each network connection is calculated based on the amount of time that will elapse before an active network connection is usable for a subsequent client request; and
 - 14 a network connection manager selecting the network connection from the proxy server to the origin server with a substantially highest service availability and a substantially lowest time-to-idle and forwarding each request from the proxy server to the origin server using the selected network connection.
- 1 2. (currently amended): A system according to Claim 1, further comprising:
 - 3 the network connection manager selecting a network connection from the proxy server not actively sending a request with a zero time-to-idle and not subject to a slow start overhead incurred responsive to flow control imposed by the sending client.

Response to Final Office Action
Docket No. 011.0201.US.UTL

1 3. (currently amended): A system according to Claim 2, further
2 comprising:

3 the network connection manager selecting a network connection from the
4 proxy server actively sending a request with a time-to-idle less than the slow start
5 overhead, plus request transfer time if the network connection is pipelined allows
6 request pipelining.

1 4. (previously presented): A system according to Claim 3, further
2 comprising:

3 the network connection manager selecting a network connection not
4 actively sending a request with a zero time-to-idle and subject to the slow start
5 overhead.

1 5. (currently amended): A system according to Claim 4, further
2 comprising:

3 the network connection manager selecting a network connection from the
4 proxy server actively sending a request with a time-to-idle less than a network
5 connection setup overhead, plus request transfer time if the network connection is
6 pipelined allows request pipelining.

1 6. (currently amended): A system according to Claim 5, further
2 comprising:

3 the network connection manager selecting a new network connection from
4 the proxy server in the absence of an existing network connection with a time-to-
5 idle less than the network connection setup overhead.

1 7. (currently amended): A system according to Claim 5, further
2 comprising:

3 the network connection manager selecting an existing network connection
4 from the proxy server with the substantially lowest time-to-idle.

Response to Final Office Action
Docket No. 011.0201.US.UTL

1 8. (currently amended): A system according to Claim 1, wherein the
2 distributed operating environment is TCP/IP-compliant, the system further
3 comprising:

4 the time estimates generator providing time estimates for each network
5 connection from the proxy server comprising at least one of TCP overhead, time-
6 to-idle, idle time, and request transfer time, wherein idle time for each network
7 connection is measured as an amount of time that has elapsed during which no
8 network traffic is being sent or received over a network connection.

1 9. (previously presented): A system according to Claim 8, the
2 network connection setup overhead comprises TCP overhead, the system further
3 comprising:

4 the time estimates generator calculating the TCP overhead by adding a
5 three-way handshake overhead to a slow start overhead.

1 10. (previously presented): A system according to Claim 8, further
2 comprising:

3 the time estimates generator calculating the request transfer time by
4 multiplying the size of the request by an average network connection speed for
5 the origin server.

1 11. (previously presented): A system according to Claim 8, further
2 comprising:

3 the time estimates generator calculating the time-to-idle upon each receipt
4 of a request by adding the time-to-idle to the product of an average network
5 connection speed for the origin server multiplied by the sum of the request size
6 and an estimated response size.

1 12. (previously presented): A system according to Claim 8, further
2 comprising:

Response to Final Office Action
Docket No. 011.0201.US.UTL

3 the time estimates generator calculating the time-to-idle upon writing data
4 to a socket by subtracting the time-to-idle from the product of an average network
5 connection speed for the origin server multiplied by the amount of data written.

1 13. (previously presented): A system according to Claim 8, further
2 comprising:

3 the time estimates generator calculating the time-to-idle upon reading data
4 from a socket, prior to header data, by subtracting the time-to-idle from the
5 product of an average network connection speed for the origin server multiplied
6 by the amount of data read.

1 14. (currently amended): A system according to Claim 1, further
2 comprising:

3 [[a]] one such proxy server configured in a location comprising at least
4 one of local to the sending clients, in the infrastructure of the distributed
5 computing environment, and local to the origin server.

1 15. (currently amended): A method for efficiently forwarding client
2 requests in a distributed computing environment, comprising:

3 receiving a plurality of non proxiable client requests commonly addressed
4 routed for forwarding to an origin server from individual sending clients into a
5 proxy server connectively interposed between the sending clients and the origin
6 server;

7 dynamically generating at the proxy server, concurrent to and during
8 processing of each request by the proxy server, time estimates of service
9 availability based on a time-to-idle for sending the requests over each of a
10 plurality of network connections from the proxy server to the origin server,
11 wherein time-to-idle for each network connection is calculated based on the
12 amount of time that will elapse before an active network connection is usable for
13 a subsequent client request; and

14 selecting the network connection from the proxy server to the origin server
15 with a substantially highest service availability and a substantially lowest time-to-

Response to Final Office Action
Docket No. 011.0201.US.UTL

16 idle and forwarding each request from the proxy server to the origin server using
17 the selected network connection.

1 16. (currently amended): A method according to Claim 15, further
2 comprising:

3 selecting a network connection from the proxy server not actively sending
4 a request with a zero time-to-idle and not subject to a slow start overhead incurred
5 responsive to flow control imposed by the sending client.

1 17. (currently amended): A method according to Claim 16, further
2 comprising:

3 selecting a network connection from the proxy server actively sending a
4 request with a time-to-idle less than the slow start overhead, plus request transfer
5 time if the network connection is pipelined allows request pipelining.

1 18. (previously presented): A method according to Claim 17, further
2 comprising:

3 selecting a network connection not actively sending a request with a zero
4 time-to-idle and subject to the slow start overhead.

1 19. (currently amended): A method according to Claim 18, further
2 comprising:

3 selecting a network connection from the proxy server actively sending a
4 request with a time-to-idle less than a network connection setup overhead, plus
5 request transfer time if the network connection is pipelined allows request
6 pipelining.

1 20. (currently amended): A method according to Claim 19, further
2 comprising:

3 selecting a new network connection from the proxy server in the absence
4 of an existing network connection with a time-to-idle less than the network
5 connection setup overhead.

Response to Final Office Action
Docket No. 011.0201.US.UTL

1 21. (currently amended): A method according to Claim 19, further
2 comprising:

3 selecting an existing network connection with from the proxy server the
4 substantially lowest time-to-idle.

1 22. (currently amended): A method according to Claim 15, wherein the
2 distributed operating environment is TCP/IP-compliant, the method further
3 comprising:

4 providing time estimates for each network connection from the proxy
5 server comprising at least one of TCP overhead, time-to-idle, idle time, and
6 request transfer time, wherein idle time for each network connection is measured
7 as an amount of time that has elapsed during which no network traffic is being
8 sent or received over a network connection.

1 23. (previously presented): A method according to Claim 22, the
2 network connection setup overhead comprises TCP overhead, the method further
3 comprising:

4 calculating the TCP overhead by adding a three-way handshake overhead
5 to a slow start overhead.

1 24. (previously presented): A method according to Claim 22, further
2 comprising:

3 calculating the request transfer time by multiplying the size of the request
4 by an average network connection speed for the origin server.

1 25. (previously presented): A method according to Claim 22, further
2 comprising:

3 calculating the time-to-idle upon each receipt of a request by adding the
4 time-to-idle to the product of an average network connection speed for the origin
5 server multiplied by the sum of the request size and an estimated response size.

Response to Final Office Action
Docket No. 011.0201.US.UTL

1 26. (previously presented): A method according to Claim 22, further
2 comprising:
3 calculating the time-to-idle upon writing data to a socket by subtracting
4 the time-to-idle from the product of an average network connection speed for the
5 origin server multiplied by the amount of data written.

1 27. (previously presented): A method according to Claim 22, further
2 comprising:
3 calculating the time-to-idle upon reading data from a socket, prior to
4 header data, by subtracting the time-to-idle from the product of an average
5 network connection speed for the origin server multiplied by the amount of data
6 read.

1 28. (currently amended): A method according to Claim 15, further
2 comprising:
3 providing [[a]] one such proxy server configured in a location comprising
4 at least one of local to the sending clients, in the infrastructure of the distributed
5 computing environment, and local to the origin server.

1 29. (original): A computer-readable storage medium holding code for
2 performing the method according to Claim 15.

1 30. (currently amended): A system for efficiently forwarding client
2 requests from a proxy server in a TCP/IP computing environment, comprising:
3 means for receiving on a proxy server connectively interposed between the
4 sending clients and an origin server a plurality of transient requests from
5 individual sending clients, each request being commonly addressed routed for
6 forwarding to [[an]] the origin server;
7 means for dynamically calculating at the proxy server, concurrent to
8 receiving and during processing of each request, time estimates of TCP overhead,
9 slow start overhead, time-to-idle, and request transfer time for sending the
10 requests over each of a plurality of managed network connections from the proxy

Response to Final Office Action
Docket No. 011.0201.US.UTL

11 server to the origin server, wherein time-to-idle for each network connection is
12 calculated based on the amount of time that will elapse before an active network
13 connection is usable for a subsequent request;

14 means for choosing the managed network connection from the proxy
15 server selected from, in order of preferred selection, a warm idle network
16 connection, an active network connection with a time-to-idle less than a slow start
17 overhead, a cold idle network connection, an active network connection with a
18 time-to-idle less than a TCP overhead, a new managed network connection, and
19 an existing managed network connection with a smallest time-to-idle; and

20 means for forwarding each request from the proxy server to the origin
21 server over the selected managed network connection.

1 31. (currently amended): A system according to Claim 30, further
2 comprising:

3 means for adding the request transfer time during each active network
4 connection selection if the managed network connection is pipelined from the
5 proxy server allows request pipelining.

1 32. (previously presented): A system according to Claim 30, further
2 comprising:

3 means for calculating the TCP overhead by adding a three-way handshake
4 overhead to a slow start overhead;

5 means for calculating the request transfer time by multiplying the size of
6 the request by an average managed network connection speed for the origin
7 server; and

8 means for calculating the time-to-idle, comprising:

9 upon each receipt of a request, means for adding the time-to-idle to
10 the product of an average managed network connection speed for the origin server
11 multiplied by the sum of the request size and an estimated response size;

Response to Final Office Action
Docket No. 011.0201.US.UTL

12 upon writing data to a socket, means for subtracting the time-to-
13 idle from the product of an average managed network connection speed for the
14 origin server multiplied by the amount of data written; and

15 upon reading data from a socket, prior to header data, means for
16 subtracting the time-to-idle from the product of an average managed network
17 connection speed for the origin server multiplied by the amount of data read.

1 33. (original): A system according to Claim 30, wherein each transient
2 request is communicated in accordance with HTTP.

1 34. (currently amended): A method for efficiently forwarding client
2 requests from a proxy server in a TCP/IP computing environment, comprising:
3 receiving a plurality of transient requests from individual sending clients
4 into a request queue on a proxy server connectively interposed between the
5 sending clients and an origin server, each request being commonly addressed
6 routed for forwarding to [[ea]] the origin server;

7 dynamically calculating at the proxy server, concurrent to receiving and
8 during processing of each request, time estimates of TCP overhead, slow start
9 overhead, time-to-idle, and request transfer time for sending the requests over
10 each of a plurality of managed network connections from the proxy server to the
11 origin server, wherein time-to-idle for each network connection is calculated
12 based on the amount of time that will elapse before an active network connection
13 is usable for a subsequent request;

14 choosing the managed network connection from the proxy server selected
15 from, in order of preferred selection, a warm idle network connection, an active
16 network connection with a time-to-idle less than a slow start overhead, a cold idle
17 network connection, an active network connection with a time-to-idle less than a
18 TCP overhead, a new managed network connection, and an existing managed
19 network connection with a smallest time-to-idle; and
20 forwarding each request from the proxy server to the origin server over the
21 selected managed network connection.

Response to Final Office Action
Docket No. 011.0201.US.UTL

1 35. (currently amended): A method according to Claim 34, further
2 comprising:

3 adding the request transfer time during each active network connection
4 selection if the managed network connection is pipelined from the proxy server
5 allows request pipelining.

1 36. (previously presented): A method according to Claim 34, further
2 comprising:

3 calculating the TCP overhead by adding a three-way handshake overhead
4 to a slow start overhead;

5 calculating the request transfer time by multiplying the size of the request
6 by an average managed network connection speed for the origin server; and

7 calculating the time-to-idle, comprising:

8 upon each receipt of a request, adding the time-to-idle to the
9 product of an average managed network connection speed for the origin server
10 multiplied by the sum of the request size and an estimated response size;

11 upon writing data to a socket, subtracting the time-to-idle from the
12 product of an average managed network connection speed for the origin server
13 multiplied by the amount of data written; and

14 upon reading data from a socket, prior to header data, subtracting
15 the time-to-idle from the product of an average managed network connection
16 speed for the origin server multiplied by the amount of data read.

1 37. (original): A method according to Claim 34, wherein each transient
2 request is communicated in accordance with HTTP.

1 38. (original): A computer-readable storage medium holding code for
2 performing the method according to Claim 34.